

## Contents

<b>1 Routine/Function Prologues</b>	<b>2</b>
1.1 Fortran: Module Interface agrmetdomain_module.F90 (Source File: agrmetdomain_module.F90) . . . . .	2
1.1.1 defnatagrmet.F90 (Source File: agrmetdomain_module.F90) . . . . .	2
1.1.2 allocate_agr_ip (Source File: agrmetdomain_module.F90) . . . . .	3
1.1.3 def_agr_ip_input (Source File: agrmetdomain_module.F90) . . . . .	3

## 1 Routine/Function Prologues

### 1.1 Fortran: Module Interface agrmetdomain\_module.F90 (Source File: agrmetdomain\_module.F90)

Contains routines and variables that define theb native domain for AGRMET observed radiation forcing

**INTERFACE:**

```
module agrmetdomain_module
```

**USES:**

```
use agrmetdrv_module
```

---

#### 1.1.1 defnatagmet.F90 (Source File: agrmetdomain\_module.F90)

Defines the kgds array describing the native forcing resolution for AGRMET data.

**REVISION HISTORY:**

11Dec2003: Sujay Kumar; Initial Specification

**INTERFACE:**

```
subroutine defnatagmet()
```

**USES:**

```
use lisdrv_module, only :lis
use lis_indices_module
implicit none
```

**ARGUMENTS:**

```
integer :: kgdsi(200)
```

**CONTENTS:**

```
call readagmetcrd(agmetdrv)
kgdsi = 0
kgdsi(1) = 0
kgdsi(2) = 1440
kgdsi(3) = 600
kgdsi(4) = -59875
kgdsi(5) = -179875
kgdsi(6) = 128
kgdsi(7) = 89875
kgdsi(8) = 179875
kgdsi(9) = 250
```

---

```

kgdsi(10) = 250
kgdsi(11) = 64
kgdsi(20) = 255
call allocate_agr_ip(lis_nc_working*lis_nr_working)
call def_agr_ip_input(kgdsi)

```

---

### **1.1.2 allocate\_agr\_ip (Source File: agrmetdomain\_module.F90)**

Allocate memory for AGRMET interpolation variables

INTERFACE:

```
subroutine allocate_agr_ip(N)
```

CONTENTS:

```

allocate(rlat(N))
allocate(rlon(N))
allocate(N11(N))
allocate(N12(N))
allocate(N21(N))
allocate(N22(N))
allocate(w11(N))
allocate(w12(N))
allocate(w21(N))
allocate(w22(N))
mo = n
nn = n
w11 = 0.0
w12 = 0.0
w21 = 0.0
w22 = 0.0

```

---

### **1.1.3 def\_agr\_ip\_input (Source File: agrmetdomain\_module.F90)**

Calculates weights and neighbor information required for AGRMET interpolation

INTERFACE:

```
subroutine def_agr_ip_input (kgds)
```

USES:

```

use spmdMod
use lisdrv_module, only:lis
use lis_indices_module

```

## CONTENTS:

```

!-----
! Calls the routines to decode the grid description and
! calculates the weights and neighbor information to perform
! spatial interpolation. This routine eliminates the need to
! compute these weights repeatedly during interpolation.
!-----

#if ( ! defined OPENDAP )
  if(masterproc) then
#endif
  kgdso = lis%d%kgds
  mo = lis_nc_working*lis_nr_working
  if(kgdso(1).ge.0) then
    call gdswiz(kgdso, 0,mo,fill,xpts,ypts,rlon,rlat,nn,0)
  endif

  call gdswiz(kgds,-1,nn,fill,xpts,ypts,rlon,rlat,nv,0)
  do n=1,nn
    xi=xpts(n)
    yi=ypts(n)
    if(xi.ne.fill.and.yi.ne.fill) then
      i1=xi
      i2=i1+1
      j1=yi
      j2=j1+1
      xf=xi-i1
      yf=yi-j1
      n11(n)=ijkgds(i1,j1,kgds)
      n21(n)=ijkgds(i2,j1,kgds)
      n12(n)=ijkgds(i1,j2,kgds)
      n22(n)=ijkgds(i2,j2,kgds)
      if(min(n11(n),n21(n),n12(n),n22(n)).gt.0) then
        w11(n)=(1-xf)*(1-yf)
        w21(n)=xf*(1-yf)
        w12(n)=(1-xf)*yf
        w22(n)=xf*yf
      else
        n11(n)=0
        n21(n)=0
        n12(n)=0
        n22(n)=0
      endif
    else
      n11(n)=0
      n21(n)=0
      n12(n)=0
      n22(n)=0
    endif
  else
    n11(n)=0
    n21(n)=0
    n12(n)=0
    n22(n)=0
  endif
end

```

```
        endif
    enddo
    mi = 864000
#if ( ! defined OPENDAP )
    endif
#endiff
```